

FINAL
IN-06-CR
4 CIT
49983
P-22

**Operator Modeling in Commercial Aviation:
Cognitive Models, Intelligent Displays, and Pilot's Assistants**

**NASA grant
(NASA Ames control number NCC 2-675; 90-55)**

Final Report

July 1, 1990 through April 30, 1994

**T. Govindaraj
C. M. Mitchell
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2300; fax (404) 894-2301
E24-613**

(NASA-CR-198609) OPERATOR MODELING
IN COMMERCIAL AVIATION: COGNITIVE
MODELS, INTELLIGENT DISPLAYS, AND
PILOT'S ASSISTANTS Final Report, 1
Jul. 1990 - 30 Apr. 1994 (Georgia
Inst. of Tech.) 22 p

N95-28203

Unclass

G3/06 0049983

Background

One of the goals of the national Aviation Safety/Automation program is to address the issue of human-centered automation in the cockpit. Human-centered automation is automation that, in the cockpit, enhances or assists the crew rather than replacing them. The Georgia Tech research program focused on this general theme, with emphasis on designing a computer-based pilot's assistant, intelligent (i.e., context-sensitive) displays, and an intelligent tutoring system for understanding and operating the autoflight system. In particular, the aids and displays were designed to enhance the crew's situational awareness of the current state of the automated flight systems and to assist the crew in coordinating the autoflight system resources.

This grant funded four separate activities: an OFMspert to understand pilot navigation activities in a 727 class (i.e., 'non-glass') aircraft; an extension of OFMspert to understand mode control in a glass cockpit, Georgia Tech Crew Activity Tracking System (GT-CATS); the design of a training system to teach pilots about the vertical navigation portion of the flight management system--VNAV Tutor, and a proof-of-concept display, using existing display technology, to facilitate mode awareness, particularly in situations in which controlled flight into terrain (CFIT) is a potential.

OFMspert for Navigation in the Non-Glass Cockpit (Serena Smith Verfurth)

Background

OFMspert for the 727 was the first component of the research funded by this grant. The current grant built on research funded at Georgia Tech under a previous Ames grant. The previous grant helped to fund the development of OFMspert (operator function model expert system).

OFMspert is both a theory and model of the use of machine intelligence to aid the human operator. Essentially OFMspert builds a real-time model of operator activity in the context of unfolding system events--it attempts to understand operator(s) actions. Given this understanding, OFMspert can then offer reminders, advice, and assistance. OFMspert can be considered one instance of human-centered automation--automation that augments operator skills rather than replacing them.

The heart of an OFMspert is an operator function model (OFM). An OFM represents how a well-trained, well motivated operator coordinates control and monitoring activities in real time. It represents how an operator decomposes (i.e., extracts features from) and represents external events, acts those representations to create new information and produces behavior from the representations and processes. The operator function model is a hierarchical decomposition that is goal-oriented at the top and event/data-driven at the bottom.

The previous OFMspert/OFM research was conducted in the context of NASA satellite ground control systems. One of the objectives of the research conducted under the sponsorship of this grant was to explore the extent to which OFMspert/OFM could be usefully generalized and applied to the aviation domain.

An Operator Function Model of Boeing 727 Navigation

As the first test of its generalizability, OFMspert was applied to data from the B-727 flight simulator at NASA Ames Research Center. In a B-727 the crew performs both manual and cognitive actions while controlling flight. Manual actions include changing radio frequencies, adjusting autopilot settings, and maneuvering power using the throttles. Cognitive actions include monitoring crucial displays within the cockpit, as well as listening and responding to verbal commands and requests from both Air Traffic Control (ATC) and other crew members in the cockpit.

The B-727 OFM represents the joint pilot-autopilot control of the B-727, specifically the activities of the pilot-flying and the pilot-not-flying. While a three member crew flies the B-727, consisting of a pilot-flying, a pilot-not-flying, and a flight engineer, the role of the flight engineer is not included in the model. In the B-727, the flight engineer's role consists primarily of executing checklists and monitoring tasks. The flight engineer, therefore, is not involved directly in the control of flight. In more sophisticated civil aviation aircraft the role of the flight engineer has been eliminated. Therefore the role of the flight engineer is not included in the model.

The B-727 operator function model has five phases of flight: Cruise, Cruise Descent Transition, Descent, Descent Approach Transition, and Approach. Six functions comprise the function level of this OFM network. There are three pilot flying functions: *Monitor/Modify Lateral Track*, *Monitor/Modify Vertical Profile*, and *Reconfigure/Check Aircraft*. There are three PNF functions: *Monitor Lateral Track*, *Perform ATC Communications*, and *Reconfigure/Check Aircraft*. The function level, which is the highest level of description in the OFM, represents pilot function and purpose in the context of the entire pilot-aircraft system. For example, the functions in this OFM are of comparable priority and take place concurrently. The pilot-flying is involved in Monitoring/Modifying the Lateral Track as well as Monitoring/Modifying the Vertical Profile continuously during Cruise to Landing. In addition, during the Approach phase, the pilot-not-flying is also involved in decisions on when to Reconfigure and Check the Aircraft. The pilot flying from Cruise to Landing has the functions of Monitoring the Lateral Track, and Performing ATC Communications. Just as the pilot-flying, the pilot-not-flying also has the additional function of Reconfigure/Check the Aircraft during the Approach phase. Figure 1 depicts one piece of the OFM: Monitor/Modify Lateral Track for the pilot flying.

OFMspert for the B-727

Figure 2 depicts a snapshot of the OFMspert Backboard, ACTIN (actions interpreter), that performs the inferencing function. Top portion is the blackboard representation; the bottom is the state space which includes actual and desired state variables derived from the flight plan and ATC clearances.

Figures 2, 3, and 4 illustrate OFMspert's operation. Depicted in Figure 2, at 509 seconds from the beginning of the flight scenario, the aircraft is more than 60 miles from Denver on the J56 radial at a cruising altitude of 31,000 feet. In Figure 2, the pilot-flying is maintaining an airspeed of 255 knots as requested by ATC. The pilot-flying's navigation radio (number 1) is tuned to 112.8 MHz (Gill). The pilot-not-flying's navigation radio is tuned to 117.0 (Denver). At 529 second, the pilot-flying announces top-of-descent. The ACTIN update is depicted in Figure 3. At 530 seconds, the pilot-flying retards the throttles to idle to descend the aircraft to 11,000 feet. The *retard throttles* action is posted on the lowest level of ACTIN, the actions level, and depicted in the figure with the circle labeled A. At 533 second the pilot-flying manipulates the rate-of-descent knob to adjust the pitch to maintain airspeed during descent. This action is connected to the *change airspeed/pitch* task node in Figure 3. In addition, the desired altitude has been updated to 11,000 feet (Circle B).

Validation

Two expert pilots were asked to indicate the intent of each action while observing the B727 OFMspert ACTIN interface. A total of 106 actions were reviewed by each pilot. The experts found only nine incorrect action connections (i.e., OFMspert inferences to be in error). Pilot reaction to OFMspert's ability to correctly interpret actions was quite positive.

Conclusion

B-727 OFMspert was a very interesting project. It was the first step in generalizing the OFMspert architecture and applying it in an aviation application. Overall the B-727 OFMspert intent inferencing ability matched human interpretations of pilot actions quite well and can be considered a valid model of crew activities for the subset of activities included in the model. The major limitation of the B-727 OFMspert is that it was most effective for modeling discrete pilot activities. Initial analysis of Ames simulator data suggested that those flights with extensive auto pilot use were most suitable and interesting to model. OFMspert, for example, does not represent continuous activities such as 'hand-flying'. Post-hoc analysis suggested that OFMspert might better be applied in a 'glass' environment. Thus B-727 OFMspert was a successful first step in pilot activity tracking and set the stage for a follow-on project that specifically addressed OFMspert, automation and mode awareness for the glass cockpit.

The B-727 OFMspert formed part of Serena Connor Verfurth M.S. Thesis. A copy appears in Appendix A. A conference proceeding summarizing the research appears in Appendix B. A journal paper describing this effort is in preparation.

The Georgia Tech Crew Activity Tracking System (GT-CATS): Understanding How Operators Use Modes of Automation to Control Complex Dynamic Systems (Todd Callantine)

Research on an intent inferencing system for the glass cockpit has progressed significantly over the course of this grant. Advances at both the methodological and software levels have brought this research to near completion. The following will detail progress in both these areas. The methodology is described first, setting the stage for a description of the computer system in its present form. The system, called the Georgia Tech Crew Activity Tracking System (GT-CATS), applies the methodology to understanding the activities performed by pilots as they use automated flight systems to fly a glass cockpit aircraft.

Refinements to the GT-CATS methodology began with narrowing the scope of the inferencing process to focus on the manner in which pilots use automation in the control of the aircraft. Glass cockpit automation uses modes to provide control options to the pilot. Modes enable operators to specify high level constraints on system performance, and, in a given mode, the automation controls low level performance parameters in a specific way. Each automatic mode therefore has characteristics that may lead the operator to select it in a particular situation. These characteristics include the manner in which the mode constrains system performance and the level of operator intervention required to use it.

Advanced cockpit automation incorporates a variety of modes which provide pilots with considerable latitude in managing the flight. However, the proliferation of increasingly complex modes has led to an important problem, namely, pilots can make errors in attempting to use the array of available modes. Mode errors can have serious consequences for flight safety. Indeed, human error in operating complex automated flight modes has played a role in a number of incidents and accidents during the last decade. Nonetheless, modes are useful, and they are likely to proliferate as both the airspace environment and available automation become more complex. GT-CATS research was therefore oriented on the premise that a thorough understanding of mode errors that occur with present cockpit automation is therefore an essential ingredient in understanding how to more effectively develop next-generation automation.

To address this need, the GT-CATS methodology for understanding how pilots use automated modes in the control of glass cockpit aircraft was developed. The capabilities provided by the GT-CATS methodology constitute the set of capabilities required to "track" the activities of pilots. The methodology also supplies a framework for understanding automated modes and their use. Because next-generation automation will undoubtedly also use modes to provide control options to pilots, such a framework is a useful contribution to the development process.

An important purpose of the GT-CATS methodology is to elucidate the factors that give rise to pilot mode errors. Such an understanding is viewed as crucial, particularly if future automation is to effectively incorporate modes that resist and/or tolerate mode errors. The long term aims of

GT-CATS research were therefore identified as helping designers of future automation develop better displays for managing the selection and use of modes, and serving as the source of knowledge for intelligent pilot aiding systems. GT-CATS could also serve as the source of knowledge for intelligent tutoring systems aimed at improving the training received by pilots who use complex flight deck automation.

The GT-CATS Methodology

The GT-CATS methodology was designed to track operator activities in real time to produce an 'understanding' of how operators use automated modes to control complex dynamic systems. First, it attempts to predict which level of automation the operator is likely to select, and when, to achieve a desired system state. It also predicts how and when the operator will setup, engage, monitor, and adjust the selected mode. Further, the methodology attempts to understand when the operator is expecting and monitoring an automatic mode transition, and to which mode. Finally, it provides a means for assessing whether the operator is actually performing actions appropriate for the situation.

OFM/OFMspert. The GT-CATS methodology is in many ways similar to that embodied in OFMspert, an architecture for a computer-based operator associate in the supervisory control of complex dynamic systems (Rubin et al., 1988). The understanding capabilities (i.e., intent inferencing property) of OFMspert are based the operator function modeling (OFM) methodology. The OFM provides the normative model from which plausible explanations for observed operator actions are generated given the current state of the controlled system. OFMs represent operator control functions using a heterarchical-hierarchical network of nodes. The network heterarchy represents the high-level activities that comprise the operator's control responsibilities. The hierarchy decomposes activities at the highest (i.e., function) level, into the subfunctions, tasks, and actions required to support the high level control functions (Figure 5). The OFM structure has been successfully demonstrated as effective for describing and prescribing operator activity in complex systems with advanced levels of automation (e.g., Mitchell, 1987).

In OFMspert, knowledge supplied by the OFM is used to construct and maintain a dynamic representation of operator activities. OFMspert uses the blackboard model of problem solving, in which operator functions, subfunctions, and tasks relevant to the current operating situation are posted to a blackboard data structure. Inferencing processes then operate on the blackboard. As actual operator actions are detected, they are understood by connecting them to every posted task they can plausibly support. If the tasks supported by the action support different functions, the action is also assumed to support multiple goals for the time being. OFMspert disambiguates this initial understanding by assessing the action's blackboard connections opportunistically when adequate information becomes available. OFMspert's intent inferencing capabilities have also been validated (Jones et al., 1990).

GT-CATS: Enhancing OFMspert for Systems with Automatic Control Modes. GT-CATS research enhances the OFM/OFMspert methodology to make it suitable for understanding specifically how operators use automatic modes to control complex dynamic systems. As with the OFM/OFMspert methodology, the GT-CATS methodology consists of two components. The first component is an explicit, task-analytic model based on the OFM that normatively specifies how operators use automatic modes to achieve the desired performance from the controlled system. The second component is a mechanism for using the model to understand operator mode selection and operation in real time.

GT-CATS' model is designed to impart an explicit automatic mode orientation to the OFM. Called an OFM for systems with automatic control modes (OFM-ACM), the model specifies normative conditions for undertaking a given activity that reflect standard operating procedures. The model structure, however, is canonical, meaning it describes all the plausible trajectories that may be followed in the course of the operator-automation interaction. This is especially important in systems with automatic control modes, because the operator can select any one of several modes to perform a particular function. When used to predict which mode(s) an operator will select, the normative conditions for selecting a mode are referenced; every possible mode that an operator may select has normative conditions for its use, and every possible mode selection is represented in the OFM-ACM. However, if the operator chooses a mode selection that is not the norm for the current operating situation, GT-CATS uses only the OFM-ACM's canonical structure (i.e., the representation of all feasible choices) to determine that the selected mode, while not the most likely, indeed supports the required function and is therefore valid.

Like the OFM, the OFM-ACM is structured as a heterarchical-hierarchical network of nodes that represent operator activities at each relevant level of abstraction (Figure 6). In the hierarchical dimension, the OFM-ACM decomposes operator functions that must be performed to meet a operational goal into the modes that can be used to perform them, and in turn decomposes each mode into the tasks, subtasks, and actions required to use it depending on the situation. (As with the OFM, such a decomposition is referred to generally as an "activity tree.") The OFM-ACM's structure is also heterarchical, because multiple functions are typically performed concurrently, and because the use of a particular mode often allows or requires supporting tasks or subtasks to be performed concurrently.

The OFM-ACM also enhances the OFM heterarchy by including an explicit hierarchical decomposition of operator activities for each phase of system control. This enables operator control responsibilities to be represented explicitly in systems whose operation is generally thought of as consisting of several phases, each of which requires operators to address a particular set of control functions. In essence, the OFM-ACM structure includes an OFM activity

tree for each relevant phase of system operation. This enhancement allows the nuances involved with using a given mode to perform a required function in a given phase to be explicated.

GT-CATS uses its OFM-ACM to understand operator mode selection and operation, just as OFMspert relies on its OFM. However, GT-CATS' mechanism for creating and maintaining a dynamic representation of operator-automation interaction is a departure from the blackboard model of problem solving used in OFMspert. In GT-CATS, the representation is maintained by matching the conditions contained in an instantiation of the OFM-ACM called the Dynamically Updated OFM-ACM (DUO). The conditions in DUO specify when a given activity should normatively be performed in the current operating situation (i.e., the state of the controlled system, the state of the control automation, and high-level goals). A simple top-down, breadth-first search through DUO is used to identify relevant activities. This procedure allows GT-CATS to establish a "best guess" as to what actions will be performed to support which mode selection, tasks, and subtasks. Establishing a best guess enables GT-CATS to immediately disambiguate the reason that an action was performed, and then wait for the action that should be performed next.

A second aspect of GT-CATS' mechanism for understanding operator mode selection and operation involves comparing actual operator actions to the dynamic representation instantiated in DUO to assess the appropriateness of the actions. This comparison is critical for tracking operator activities, because detected operator actions provide the information required for confirming whether expectations about automatic mode operation have been met, or whether a mode error has occurred. If the operator performs an action that supports a postulated mode, GT-CATS explains the action accordingly. If the operator chooses a mode different from the postulated mode, GT-CATS assesses the detected action to determine if it can be explained as supporting an alternative valid mode. If it can, GT-CATS revises its previously postulated mode selection and begins focusing on the next action that should be performed in support of the mode the operator actually selected.

The GT-CATS Architecture

Once the GT-CATS methodology was firmly established, research efforts focused on the development of a generic architecture that embodies the methodology, and could be used to apply it understanding pilot activities in real time. The generic GT-CATS architecture is depicted in Figure 7. The architecture incorporates the elements necessary to implement the GT-CATS methodology for tracking the activities of operators using automation to control a complex dynamic system. Seven important functional components are shown: the GT-CATS interface, the high level controller (event queue), the state space and high level goal representations, the action handler, the OFM-ACM, and the Dynamically Updated OFM-ACM (DUO) with its associated search procedures.

The arrows in Figure 7 represent information flow between the individual components. The GT-CATS interface receives information about the state of the controlled system and the state of the control automation, as well as operator actions performed on the control automation, from the controlled system. The interface parses this information and sends it to the high level controller for processing. The high level controller schedules appropriate events in its event queue, and initiates processing of these events in accordance with its timing functions. These events include updates of GT-CATS' state space representation, searches of DUO that dynamically update the representation of operator activities, and action handling events. As shown in Figure 3, the action-handler also sends events required to check on actions to the high level controller for scheduling and later processing. The DUO search procedures take information from the state space representation, along with high level goal information, and use it to access the model. The representation built from the model also supplies information to the action handler for use in assessing operator actions. These elements of the generic GT-CATS architecture are now described.

GT-CATS Interface. The GT-CATS interface provides a link between the controlled system, its control automation, the control actions undertaken by the operator(s), and GT-CATS proper. Abstractly, it is simply an endless loop that parses the information it receives and sends this information to the high level controller. The information comprises that required to update the state space representation, update the dynamic representation of current operator activities encapsulated in DUO, or signal the action handler to process operator control actions.

High Level Controller. The high level controller coordinates the execution of each type of event according to the information sent from the GT-CATS interface. The high level controller includes timing functions that synchronize GT-CATS' to the controlled system in real time, an event queue, and a mechanism for scheduling and processing events in the event queue. As indicated by the arrows in Figure 7, the action handler can also send events to the high level controller for scheduling on the event queue and subsequent processing. On each processing cycle the high level controller processes all the events whose scheduled processing time is at or before the current system time, as dictated by the timing functions.

State Space Representation. The state space representation is a dynamic representation of the important state variables that characterize the state of the controlled system and control automation. New state information is received by the GT-CATS interface and scheduled for updating at the appropriate time by the high level controller. On the high level controller's next processing cycle the state space values are updated accordingly to maintain a current representation of the control situation.

High Level Goals (Performance Specifications). The representation of high level operator goals for system performance is crucial for determining which activities in DUO are currently

important. The correspondence (or lack thereof) between the high level goals and the current state space is the key indication that a particular set of control activities are relevant in a given situation.

OFM-ACM, DUO, and Associated Search Procedures. The OFM-ACM is instantiated in DUO. DUO and its associated search procedures provide the domain and processing knowledge required for understanding operator activities. The high level controller executes an event to update DUO after events that update the current state space have been executed. The search procedures draw information from the high level goals and state space representations to create a summarized description of the current operating situation which is used to reference the conditions in DUO. The search procedure identifies the activities that are relevant to the current operating situation. The set of activities in DUO that are identified as relevant constitute the set of expectations GT-CATS has about the functions, mode selections, and supporting tasks, subtasks, and actions operators should perform to achieve the desired goals. GT-CATS then attempts to confirm these hypotheses using the actual actions performed by operators.

Action Handler. The action handler gives and receives information from DUO and the high level controller in order to implement the action handling portion of the GT-CATS methodology. When the GT-CATS interface detects operator control inputs to the control automation, the high level controller schedules the event for processing by the action handler. The action handler also sends the high level controller an event to be scheduled as a result of an action becoming relevant in DUO. Detected actions that cannot be explained on the basis of normative expectations from DUO require the action handler to revise the set of activities that are relevant in DUO. Thus, double-headed arrows link the action handler to both the high level controller and DUO in Figure 3.

GT-CATS' action handler supplies the crucial activity tracking capabilities required understand the actions operators perform when using automatic modes. The action handler's first job is to deal with actions that are deemed relevant for supporting postulated mode selections. The action handler also schedules an event to check that the expected action has indeed been detected after a prescribed time interval. When this event is processed, the action handler notes either that the action was detected during the interval or that the action may be missed or late.

When an expected action corresponds to an actual action performed by a operator, GT-CATS' action handler issues an explanation for the action using DUO. Like the initial expectation of the action, the explanation is a statement to the effect that the action is explained as supporting the subtask, task, mode, and function above it in DUO (i.e., the ones it was expected to support). This explanation indicates that the operator performed the correct action as part of task involved with the normative use of a mode to perform a required function.

GT-CATS' action handler can also produce an accurate explanation for an action that was performed in support of a unexpected, but still valid, mode selection. The canonical OFM-ACM structure (i.e., the representation of all feasible mode selections) instantiated in DUO allows the validity of a selected mode to be confirmed. If the action can be explained accordingly, the action handler explains it, revising its previous expectation in the process. If not, the action handler issues a statement indicating that the action is unexplainable. Conceptually, the explanation revision process works as follows. GT-CATS' action handler first uses DUO to determine if the action does support a mode that could be used to perform the required operational function. It then determines if the mode is in fact in use. If the task and subtask that the action supports, according to DUO, are now irrelevant (i.e., the operator(s) are not addressing them, because they have already been accomplished), then the action is explained as supporting the operator's mode selection option.

Applying GT-CATS to the Glass Cockpit

Glass cockpit aircraft are characterized by advanced flight control automation that offers multiple modes for controlling the flight path of the aircraft at different levels of automation. In developing GT-CATS, some important assumptions were made about the glass cockpit domain. First, GT-CATS assumes that Air Traffic Control (ATC) clearances are received via datalink technology. Datalink effectively automates the process by which clearances are acknowledged by the pilots. Other assumptions address the contents of the OFM-ACM. These assumptions direct the focus of the OFM-ACM on the use of automation modes for controlling the aircraft. Finally, it is assumed that data about pilot control actions, important state variables, and information programmed in the automation is available, and that the automation always works as advertised. Given these assumptions, the generic GT-CATS architecture depicted in Figure 3 above was implemented for the glass cockpit. The GT-CATS architecture as implemented for aviation is shown in Figure 4 below. The GT-CATS interface and high level controller work the same way as described above for the generic GT-CATS architecture. The other elements of GT-CATS for the glass cockpit have been tailored to suit the domain.

In the glass cockpit, high level goals of the flight are dictated by the desired flight path. Because in many cases the specific flight path chosen is left to the discretion of the pilots, the desired flight path is really an "envelope" defined by ATC clearances and the performance limitations of the aircraft. In GT-CATS, the desired flight path is therefore represented by a data structure called the "limiting operating envelope." The limiting operating envelope summarizes the preplanned flight path along with dynamic deviations specified by ATC. Together, this information comprises that which is required to dynamically represent the high level goals of the pilots.

The GT-CATS state space consists not only of aircraft state variables (e.g., heading, altitude, airspeed, etc.), but also of autoflight system state variables (e.g., target values, engaged modes) and programmed information. Information about flight progress (e.g., top-of-climb point passed or not) is also included. This additional state information is essential for generating hypotheses about which modes pilots should use to change the flight path of the aircraft when such changes are required.

As in the generic architecture, the OFM-ACM is first instantiated in DUO. DUO and its associated search procedures then provide GT-CATS with knowledge required for understanding pilot activities. The search procedures draw information from the limiting operating envelope and state space representations to create a summarized description of the current operating situation which is used to reference the conditions in DUO. The search procedure identifies the activities that are relevant to the current operating situation. This set of activities in DUO constitute the set of expectations GT-CATS has about the functions, mode selections, and supporting tasks, subtasks, and actions pilots should perform to achieve the desired flight path, as specified by the limiting operating envelope. GT-CATS then attempts to confirm these hypotheses using the actual pilot control actions performed on the autoflight system of the aircraft.

GT-CATS' action handler is responsible for attempting to understand the actions pilots perform when using complex autoflight system modes. When an action is identified as relevant in DUO, the action handler first schedules an event to check that the pilot has actually performed the action within a prescribed time interval. If the pilot fails to perform the action by the time the event is processed, the action handler notes that the action may be missed or late. If the pilot performs the action expected according to DUO within the prescribed time interval, the action handler issues an explanation for the action. The last function of the action handler is to produce an explanation for a pilot action that supports an unexpected, but still valid, mode selection. For this purpose, DUO instantiates the canonical structure of the OFM-ACM (i.e., the representation of all feasible mode selections). GT-CATS' action handler then refers to DUO to process the unexpected action. If the action supports a valid mode selection, the action handler issues an explanation for it, revising its previous expectations in the process. If the action is not valid, the action handler issues a statement indicating that the action cannot be explained.

Summary

GT-CATS works by using its OFM-ACM to determine what functions pilots should perform to acquire and track the desired flight path and hypothesize which automated mode(s) will be used to perform these functions. It also hypothesizes the tasks, subtasks, and actions pilots will perform to properly use the postulated mode(s). Given these hypotheses, GT-CATS attempts to explain pilot actions. Four additional conceptual elements augment these basic elements of the methodology in the GT-CATS glass cockpit implementation, viz., a means of dynamically representing the

desired flight path (high level goals), a means of maintaining an updated state space representation in real-time, and a high level controller to perform real-time event processing.

Validation

GT-CATS research has also addressed the development of a procedure for validation of GT-CATS. The GT-CATS methodology will be validated through a series of empirical studies designed to demonstrate its use and effectiveness in understanding how pilots use complex autoflight system modes. These studies will follow the general procedure set forth by Jones et al. (1990). The validation will proceed in three phases. First, pilots acquainted with the autoflight system of the glass cockpit will serve as subjects for pilot studies in which the adequacy of the part task simulator and experimental scenarios will be assessed. After the pilot studies are completed, NASA Ames Advanced Concepts Flight Simulator data will be analyzed by GT-CATS to determine if GT-CATS can interpret the actions performed by test pilots during the NASA flight scenarios. The final phase of validation will involve using pilots as subjects flying a Boeing 757 autoflight system part-task simulator linked to GT-CATS in real time. GT-CATS' explanations for pilot actions in each experimental scenario will be compared to the pilots' own explanations to determine the degree to which GT-CATS' explanations match those of the pilots. Pilots will also be asked to provide verbal protocols of explanations for their actions which will be compared to the explanations from GT-CATS. GT-CATS' explanations should correspond closely to the pilots' explanations, if GT-CATS effectiveness is to be confirmed.

The VNAV Tutor (Ed Crowther and Alan Chappell)

One of the major tasks of pilots of modern aircraft is monitoring and understanding the status and behavior of the auto flight system, i.e., mode awareness. Control modes of the system change due to pilot commands (manually) or in response to system events (automatically). The flexible and dynamic nature of the system increases both the functionality of the control system and the cognitive demands placed on the pilot. In order to maintain mode awareness in this dynamic environment, the pilot must be continuously vigilant of indications from several locations within the cockpit. Lacking accurate and complete system knowledge and/or an interface that clearly presents the system state and constraints, the pilot may misunderstand the control modes. Pilots often cite vertical path navigation (VNAV) as a flight management system function that surprises them. The VNAV Tutor, a computer-based training system, was developed to address this issue. The VNAV Tutor attempts to improve the pilot's understanding of VNAV control modes and the interaction of the mode control panel functions with the flight management system during VNAV usage. Furthermore, it attempts to help pilots build a robust conceptual model of vertical navigation operation (Figure 8). An evaluation showed that the VNAV Tutor enhanced both the conceptual understanding and use of the vertical navigation function by pilots transitioning to

aircraft with sophisticated auto flight systems. Appendix C contains a copy a manuscript submitted for publication that describes the VNAV Tutor and associated evaluation.

3D Primary Flight Display with Terrain Information

An important world-wide aviation safety problem is still the controlled-flight-into-terrain or CFIT accident. Area navigation and onboard terrain elevation data bases offer the potential for improved cockpit displays near by terrain.

This project has developed a prototype primary flight display format designed to re-enforce the pilot's model of both lateral and vertical navigation in near-terrain situations. This new display format is referred to as the Spatial Situation Indicator (SSI). Specific emphasis has been placed on the terminal phase of flight with terrain modeling in the vicinity of the departing and destination airport.

The unique design incorporated perspective symbology which depicts a prediction of the aircraft's predicted position and terrain clearance information for up to 75 seconds ahead of the aircraft. Projection of the flight path is based on a "fast time" modeling technique described by Grunwald (1985). Traditional flight paths use the "tunnel in the sky" approach which present no reference to the ground elevation e.g., Grunwald (1982). The technique developed for this research utilized roll stabilized vertical lines "whiskers" positioned at 15 second intervals out to 75 seconds. Figure 9 illustrates the virtual "whiskers" and flight path. The whiskers are displayed in pairs of equal distant widths so that in steady level flight a perspective path is projected. The whiskers are color coded using green and yellow. The green lower portion extends from the predicted aircraft altitude at that interval to the terrain below. Its length therefore is a direct representation of the terrain clearance at that point in the aircraft's path given no changes in aircraft flight path.

The display also incorporates a dynamically color coded terrain grid. The color coding is based upon aircraft predicted height and terrain spot elevations. The color coding uses dark green for safe terrain and dark red for dangerous terrain. The terrain grid is comprised of a triangular mesh with each triangle having sides of 2 nautical miles (NM). Man-made obstructions such as radio towers are also shown on the terrain grid. Information for building the terrain and obstruction files is obtained from the approach plates for each runway in the scenario.

An experimental evaluation of the display was conducted on-site at a major U.S. airline. Experimental participants are current glass cockpit flight instructors. Each experimental subject, after training to familiarize him/herself with the part-task aircraft simulator and interface, will fly three scenarios based on actual controlled flight into or toward terrain as described by Bateman (1991).

Each experimental participant used one of the three displays: the baseline cockpit display, the primary flight display, and this display with flight path predictor and ground terrain

information. A total of eighteen pilots will participate, six with each display. Attention diverting tasks are implemented to match as closely as possible the scenarios as they are described by Bateman. ATC communications are implemented using simple voice communications without supporting electronic intercoms. The experimenter carries out the air traffic controller (ATC) communications. The goal of the experiments is to measure how quickly pilots can detect dangerous terrain with the three different display formats. Response time of the pilot for corrective action is recorded as well as MCP inputs. Data from this experiment are still being analyzed. The project formed a portion of Jim Williams' M.S. thesis which is in preparation.

Conclusion

This research encompassed numerous projects, some of which are still on-going; it also provided the foundation for new efforts. GT-CATS is being completed under a follow-on grant with NASA Ames. In addition, the VNAV Tutor provided essential insight and experience for exploring training issues and the role of computer-based tutors in aviation. On-going work greatly benefits from the VNAV Tutor development and evaluation.

This grant allowed the various members of the Center for Human-Machine Systems Research Center group at Georgia Tech to begin to become 'aviation-literate'. Although challenging at times, it has been intellectually stimulating. To some extent, the enrollment of two Delta pilots in our doctoral program attests to the quality of our initial work. We hope to contribute significantly to the basic knowledge on human-centered aviation on the flight deck.

Finally, in Appendix D, is enclosed a copy of a chapter to appear in Bill Rouse's series on Human Interaction with Complex Systems. Though not aviation-related, this chapter describes the MSOCC project in which both the operator function model and OFMspert were initially developed. Various pieces of the MSOCC project were undertaken with the financial support of NASA Ames under a previous grant and the continued intellectual stimulation of Ev Palmer

References

- Jones, P. M., Rubin, K. S., and Mitchell, C. M. (1990). Validation of intent inferencing by a model-based operator's associate. *International Journal of Man-Machine Studies*. 33, 177-202.
- Mitchell, C. M. (1987). GT-MSOCC: A research domain for modeling human-computer interaction and aiding decision making in supervisory control Systems. *IEEE Transactions Systems, Man, Cybernetics*. 17(4), 553-572.

Rubin, K.S., Jones P. M., and Mitchell, C. M. (1988). OFMspert: Inference of operator intentions in supervisory control using a blackboard architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(4), 618-637.

Appendices

(Contact Principal Investigators for Copies of the Appendices)

Appendix A

Verfurth, S. C. (1991). Modeling the Pilots and Constructing an Intent Inferencer for a Boeing 727 Cockpit, M.S. thesis, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Appendix B

Verfurth, S. C., Govindaraj, T., and Mitchell, C. M. (1991). OFMspert for the 727: An investigation into intent inferencing on the flight deck. *Proceedings of the 1991 International Conference of Systems, Man, and Cybernetics*, 1311-1316, Charlottesville, Virginia.

Appendix C

Chappell, A. R., Crowther, E. G., Mitchell, C. M., and Govindaraj, T. (1995). The VNAV Tutor: Addressing a mode awareness difficulty for pilots of the glass cockpit, submitted to the special issue on Human Interaction with Complex Systems of the *IEEE Transactions on Systems, Man, and Cybernetics*.

Appendix D

Mitchell, C. M. (1995). GT-MSOCC: Operator models, model-based displays, and intelligent aiding. In W. B. Rouse (ed.). *Human-technology interaction in complex systems*. Vol. 6, Greenwich, Connecticut: JAI Press Inc, in press.

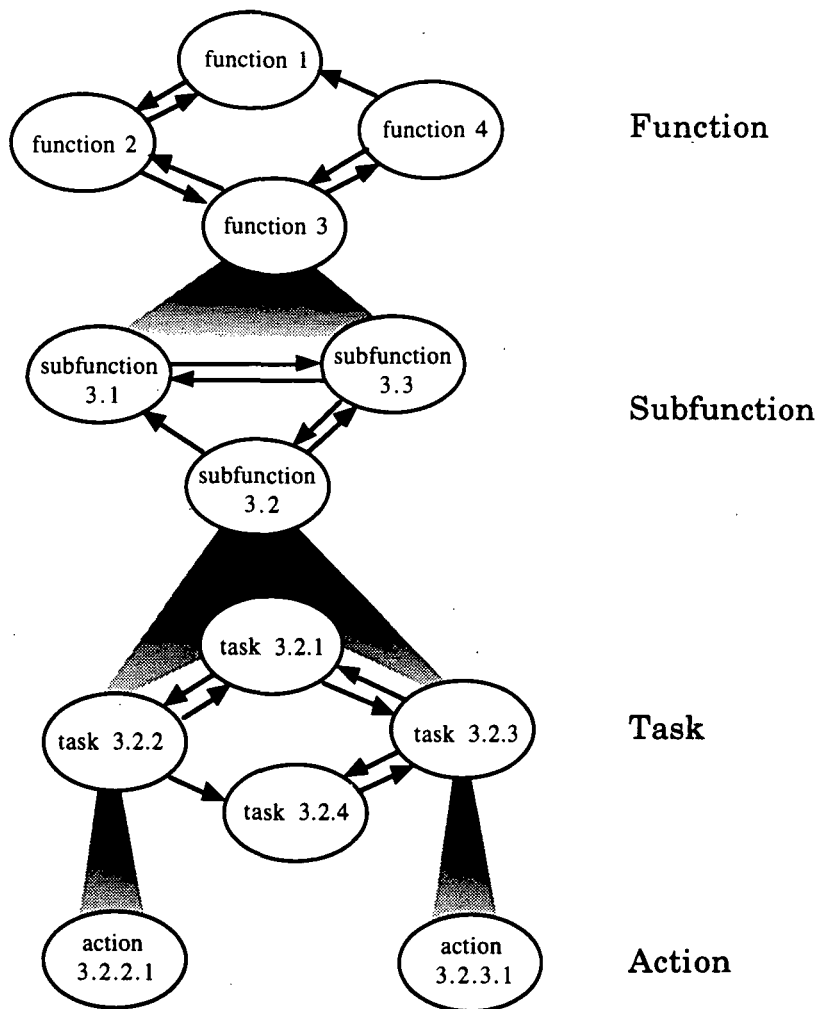


Figure 5. Generic OFM structure

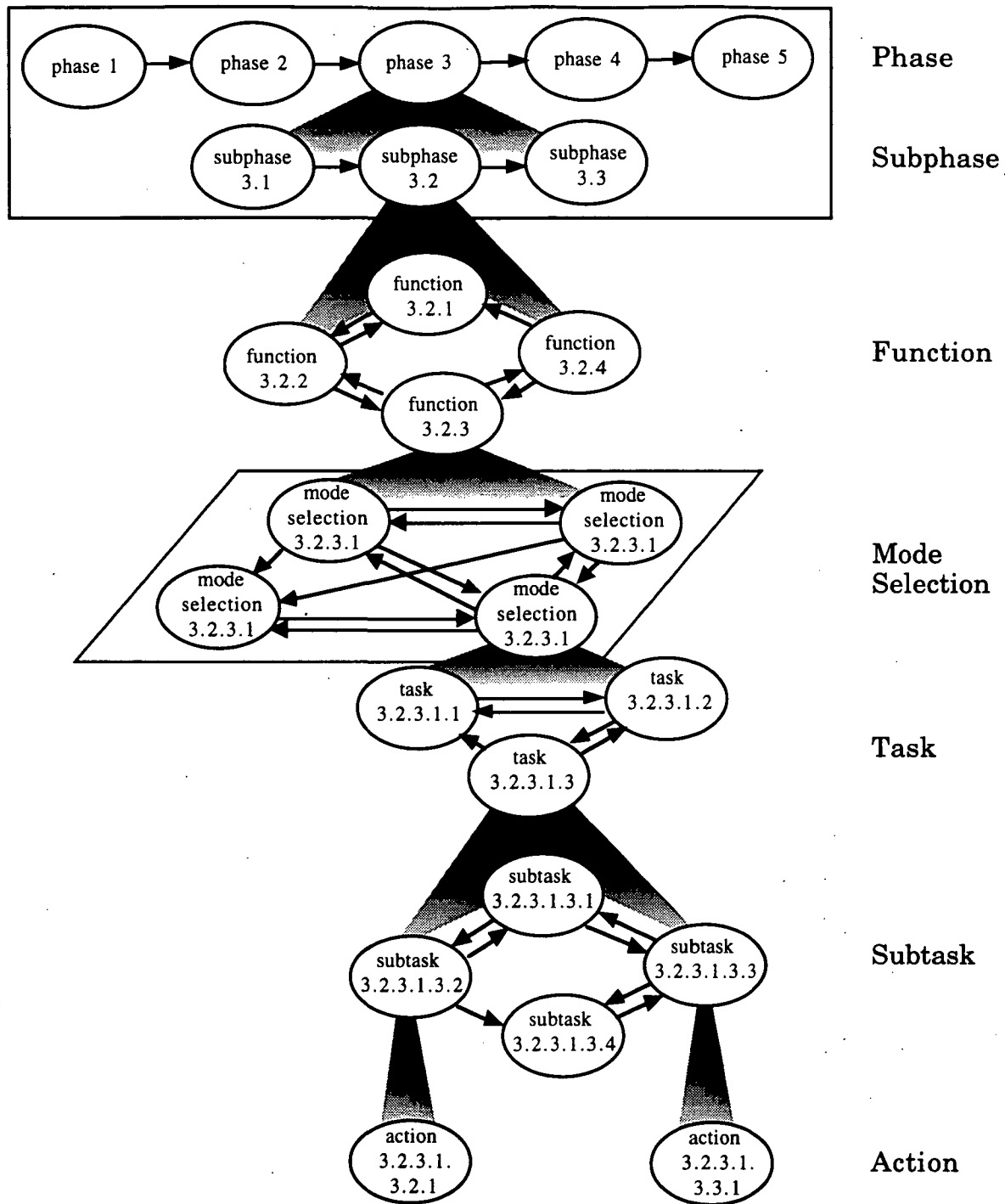


Figure 6. Generic OFM-ACM structure showing explicit phase/subphase activity decomposition and mode selection level

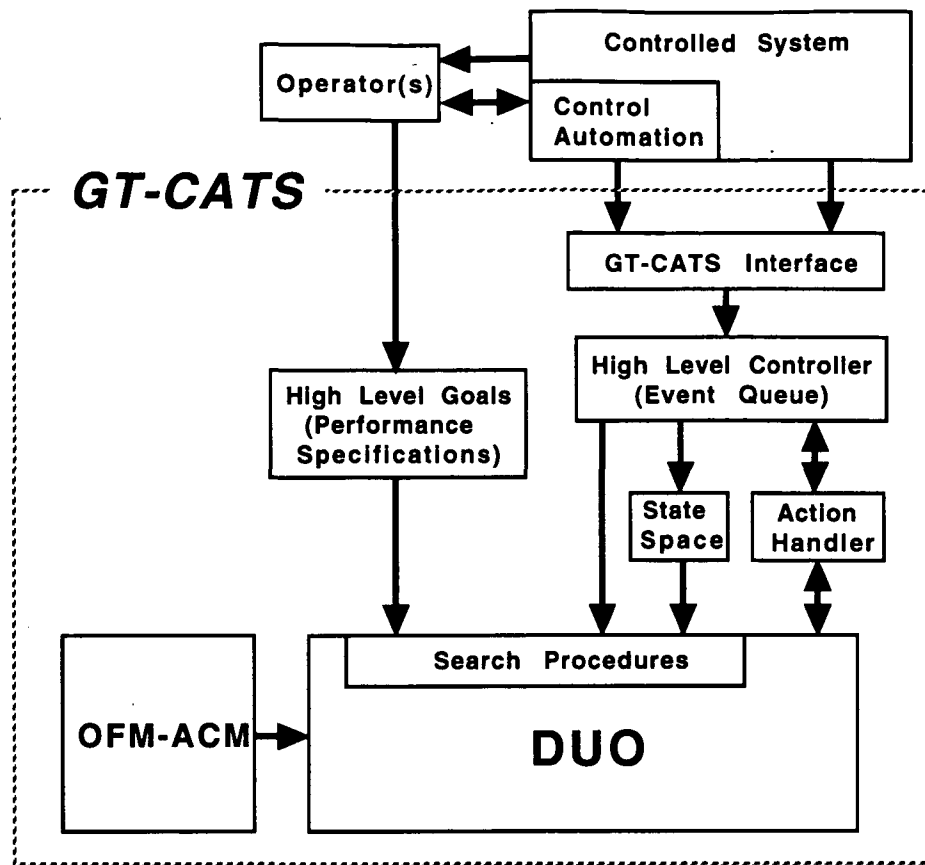


Figure 7. Generic GT-CATS Architecture

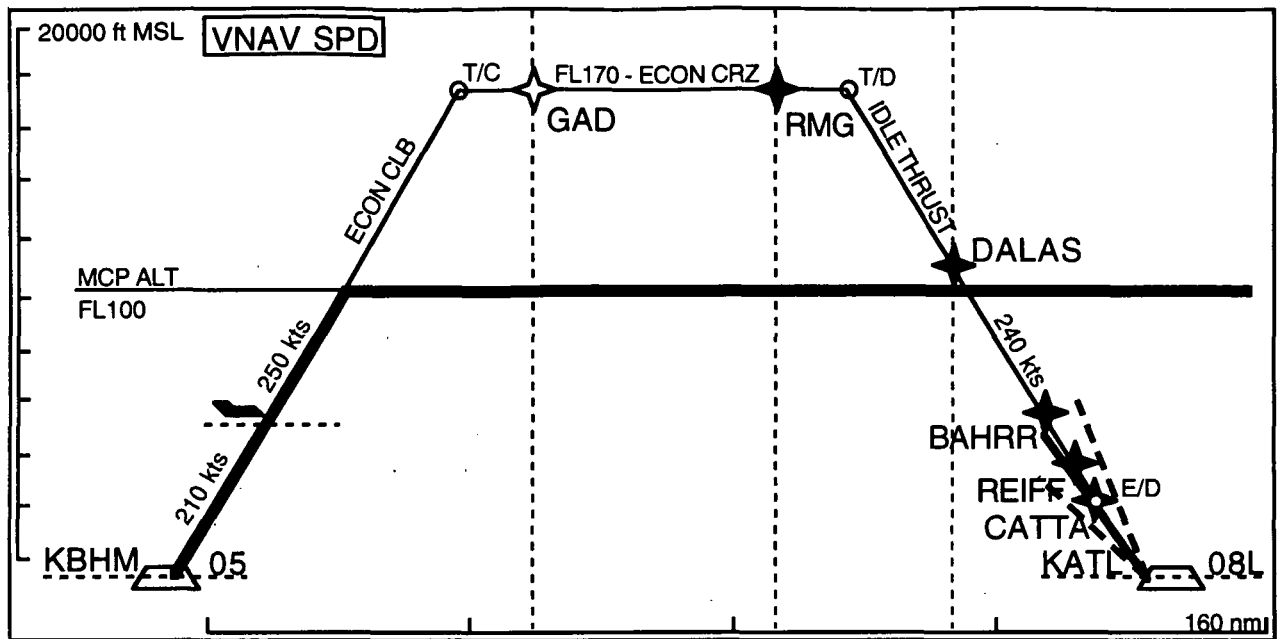


Figure 8. Vertical Profile Display: The Heart of the VNAV Tutor

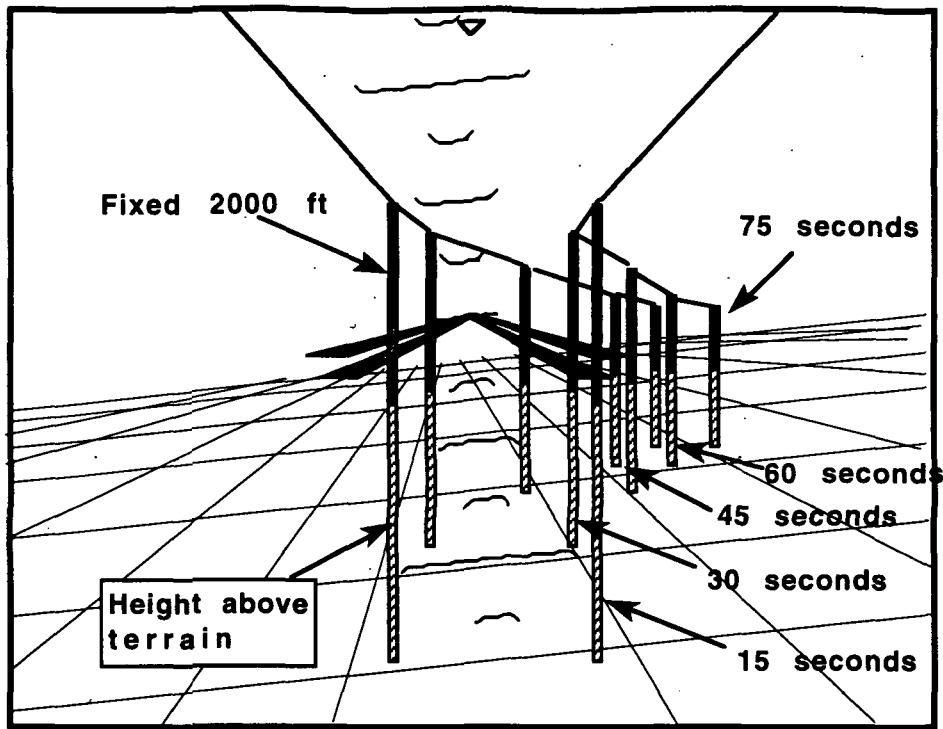


Figure 9. The SSI Display